





1. (Kubernetes)

```
# deployment.yaml   
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: laravel-app  
spec:  
  replicas: 6  
  selector:  
    matchLabels:  
      app: laravel  
  template:  
    metadata:  
      labels:  
        app: laravel  
    spec:  
      containers:  
      - name: app  
        image: your-registry/laravel-app:1.0.0  
        ports:  
        - containerPort: 9000  
        envFrom:  
        - configMapRef:  
            name: laravel-config  
      resources:  
        requests:  
          cpu: "500m"  
          memory: "512Mi"  
        limits:  
          cpu: "1000m"  
          memory: "1024Mi"  
      livenessProbe:  
        httpGet:  
          path: /health
```

```
    port: 9000
    initialDelaySeconds: 30
    periodSeconds: 10
---
# Horizontal Pod Autoscaler
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: laravel-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: laravel-app
  minReplicas: 4
  maxReplicas: 20
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 70
```

2. (Istio)

```
# 
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: laravel-vs
spec:
  hosts:
  - "yourdomain.com"
  gateways:
  - laravel-gateway
  http:
  - route:
```

```
- destination:  
  host: laravel-app  
  port:  
    number: 9000  
  weight: 100  
retries:  
  attempts: 3  
  perTryTimeout: 2s  
timeout: 5s
```



1.

```
// config/database.php  
'mysql' => [  
  'read' => [  
    'host' => [  
      env('DB_READ_HOST_1', 'read1.db.cluster'),  
      env('DB_READ_HOST_2', 'read2.db.cluster'),  
    ],  
  ],  
  'write' => [  
    'host' => env('DB_WRITE_HOST', 'write.db.cluster'),  
  ],  
  'sticky' => true,  
  'driver' => 'mysql',  
  'database' => env('DB_DATABASE', 'forge'),  
  'username' => env('DB_USERNAME', 'forge'),  
  'password' => env('DB_PASSWORD', ''),  
  // ...  
],
```

2. Redis

```
// config/database.php
'redis' => [
    'cluster' => env('REDIS_CLUSTER', true),

    'default' => [
        [
            'host' => env('REDIS_HOST_1', 'redis1'),
            'password' => env('REDIS_PASSWORD', null),
            'port' => env('REDIS_PORT', 6379),
            'database' => 0,
        ],
        [
            'host' => env('REDIS_HOST_2', 'redis2'),
            'password' => env('REDIS_PASSWORD', null),
            'port' => env('REDIS_PORT', 6379),
            'database' => 0,
        ],
    ],

    'cache' => [
        // [ ] Redis
    ],
],
```




1. []

```
// [ ]
$value = Cache::remember('key', $seconds, function () {
    return DB::table(...)->get();
});

// [ ]
Cache::tags(['people', 'authors'])->put('john', $john, $seconds);
```

2.

```
//   
'connections' => [  
  'default' => [  
    'driver' => 'redis',  
    'connection' => 'default',  
    'queue' => env('REDIS_QUEUE', 'default'),  
    'retry_after' => 90,  
    'block_for' => null,  
  ],  
  
  'processing' => [  
    'driver' => 'sqs',  
    'key' => env('AWS_ACCESS_KEY_ID'),  
    'secret' => env('AWS_SECRET_ACCESS_KEY'),  
    'prefix' => env('SQS_PREFIX'),  
    'queue' => env('SQS_QUEUE'),  
    'region' => env('AWS_DEFAULT_REGION'),  
  ],  
],
```



1. Prometheus

```
# prometheus.yml  
scrape_configs:  
  - job_name: 'laravel'  
    metrics_path: '/metrics'  
    static_configs:  
      - targets: ['laravel-app:9000']  
    relabel_configs:  
      - source_labels: [__meta_kubernetes_pod_name]  
        target_label: pod
```

2. (ELK)

```
// config/logging.php
'stacks' => [
    'elastic' => [
        'driver' => 'custom',
        'via' => \App\Logging\ElasticsearchLogger::class,
        'level' => 'debug',
        'hosts' => [
            [
                'host' => env('ELASTICSEARCH_HOST', 'elasticsearch'),
                'port' => env('ELASTICSEARCH_PORT', 9200),
                'scheme' => env('ELASTICSEARCH_SCHEME', 'https'),
            ]
        ],
    ],
],
```

CI/CD

1. GitLab CI

```
stages:
  - test
  - build
  - deploy

unit_tests:
  stage: test
  image: php:8.1
  script:
    - apt-get update && apt-get install -y git unzip
    - curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer
    - composer install --prefer-dist --no-ansi --no-interaction --no-progress --no-scripts
    - vendor/bin/phpunit
```

```

build_image:
  stage: build
  image: docker:latest
  services:
    - docker:dind
  script:
    - echo "$CI_REGISTRY_PASSWORD" | docker login -u "$CI_REGISTRY_USER" --password-stdin $CI_REGISTRY
    - docker build -t $CI_REGISTRY_IMAGE:latest .
    - docker push $CI_REGISTRY_IMAGE:latest

deploy_production:
  stage: deploy
  image: bitnami/kubectl:latest
  script:
    - kubectl apply -f kubernetes/deployment.yaml
    - kubectl rollout status deployment/laravel-app
  environment:
    name: production
    url: https://yourdomain.com
  when: manual

```



1.
 - WAF (Web Application Firewall)
 - VPC
 - DDoS

2.

```

//  HTTPS
URL::forceScheme('https');

// CSP 
header("Content-Security-Policy: default-src 'self'");

```

3.
 - Vault KMS
 -



1.
 -
 -
2.
 - ()
 -
3.

```
//   
event(new OrderShipped($order));  
  
//   
Http::post('inventory-service/api/update', [...]);
```